CSE 493V Final Project Report

Exploring Pupil Tracking

YOUSSEF BEN TALEB and ZEYNEL C. GURBUZ, University of Washington

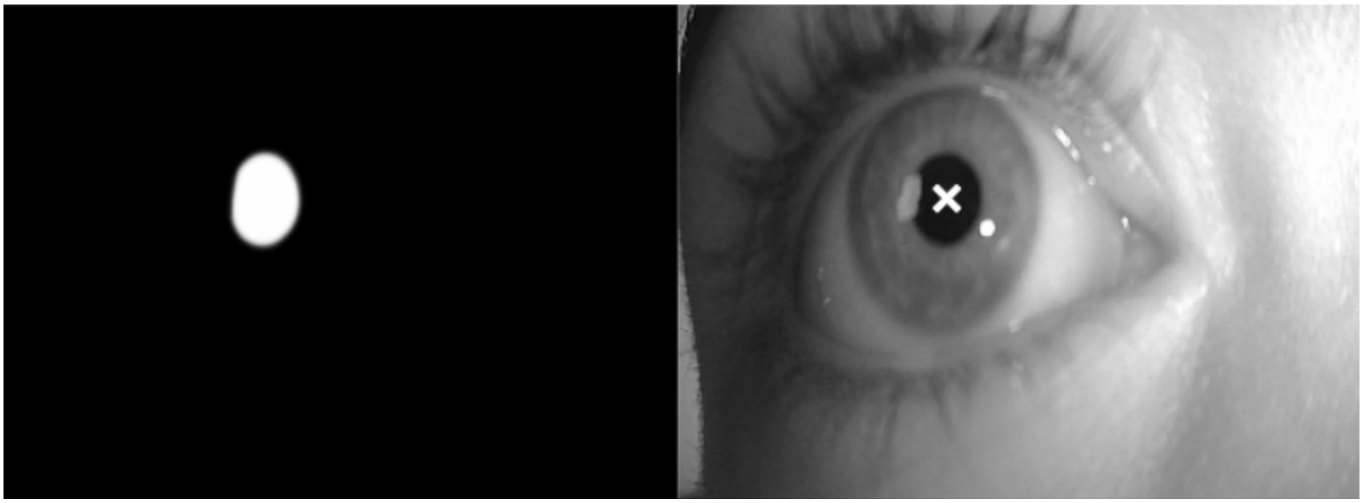

Fig. 1. This figure shows the Eye Tracking Capabilities of our Project including Pupil Detection From the right image shown on the Right Side Image, where we detect all pixels on the screen that show the user's pupil shown on the Left Side Image. We can also see a cross on the Right Side Image to demonstrate our pupil center calculations.

In this project, we present a proof of concept for a stand-alone gaze tracking system. Our system leverages advanced machine learning models and efficient computer vision techniques to accurately detect pupil positions and track gaze direction in real-time. We have not integrated our system with any specific VR headset; instead, our focus is on demonstrating the capabilities of these technologies and their potential to enhance user experiences in virtual reality environments. While our project does not pioneer new tracking technologies, it serves as a significant proof of concept, demonstrating the feasibility and utility of such systems in broader applications. We hope that our work will inspire further development and refinement in the field of gaze tracking.

1 INTRODUCTION

Eye tracking is a crucial technology that has gained significant importance in various fields, including virtual reality (VR). One of the key reasons why eye tracking is needed is its potential to enhance user experience and immersion in VR environments. By tracking the movement and gaze direction of the user's eyes, VR systems can precisely determine what the user is looking at and adjust the displayed content accordingly. This capability allows for more realistic and interactive experiences, as the virtual environment can respond dynamically to the user's visual attention.

In VR, eye tracking plays a significant role in a technique known as foveated rendering. Foveated rendering takes advantage of the fact that our eyes have higher visual acuity in the center of our field of view (fovea) compared to the peripheral vision. By tracking the user's eye movements, the VR system can identify the area within the field of view that corresponds to the fovea and allocate higher rendering resources to that region. At the same time, the system can reduce the level of detail and rendering quality in the peripheral areas. This approach optimizes computational resources and improves performance, allowing for more efficient rendering and reducing the hardware requirements for running VR applications.

To achieve gaze tracking in VR using eye tracking technology, several challenges need to be addressed. One of the primary challenges is accurately mapping the user's gaze direction to the corresponding point in the virtual environment. This involves calibrating the eye tracking system to the user's eyes and establishing a reliable correlation between eye movements and visual attention. Additionally, the VR system needs to handle potential issues such as eye drift, where the user's gaze may wander even when their eyes are fixated on a specific point. Algorithms and techniques are employed to minimize these errors and ensure accurate and reliable gaze tracking.

To address the challenges associated with eye tracking and gaze tracking in VR, we employ a combination of machine learning models and computer vision techniques to solve these problem. Although previous work has demonstrated the ability for Eye Tracking, our paper serves as a proof of concept of these abilities and the applications they can enable.

1.1 Contributions

- Advanced machine learning techniques: We demonstrate the effectiveness of advanced machine learning models for

Authors' address: Youssef Ben Taleb, yben@cs.washington.edu; Zeynel C. Gurbuz, zgurbuz@cs.washington.edu, University of Washington.

accurately detecting a user's pupil. These models leverage complex algorithms and deep learning architectures to analyze eye images and accurately identify the position and shape of the pupil.
- Efficient computer vision techniques: In addition to machine learning models, we also explore fast and efficient computer vision techniques for pupil detection. These techniques utilize image processing algorithms and feature extraction methods to achieve real-time pupil tracking, although with reduced accuracy compared to the advanced machine learning models.
- Accurate gaze tracking: We showcase the successful integration of calibration techniques and regression models to accurately track eye gaze based on the detected pupil positions. Through careful calibration, we establish a reliable mapping between eye movements and gaze direction, enabling precise tracking of the user's visual attention in virtual environments.

In summary, our project contributes not necessarily by pioneering new eye-tracking technologies but by demonstrating the implementation, integration, and application of these systems in VR experiences without the resources of tech giants.

2 RELATED WORK

Our model builds upon the advancements in eye tracking and gaze tracking techniques. Leveraging advanced machine learning and computer vision approaches, we have developed a robust and accurate system for pupil detection and gaze tracking. Our model combines the precision of state-of-the-art CNN models from DeepVOG [Kassner et al. 2014], to track the eye pupil with remarkable accuracy. This allows us to provide users with highly precise eye tracking capabilities, ensuring a more immersive and interactive VR experience.

However, we recognize the computational demands that come with employing such sophisticated models, which can pose limitations for some users. To address this challenge, we have integrated center-of-the-eye tracking techniques, inspired by Pupil Labs' edge detection approach [Yiu et al. 2019]. This enables us to offer a faster and more efficient alternative for eye tracking, ensuring a seamless experience while respecting GPU computation limits. By providing users with the flexibility to choose their preferred tracking model, we aim to cater to individual preferences and optimize the overall tracking performance for each user.

3 METHOD

This system for gaze tracking consists of a series of procedures to capture, process and analyze the user's gaze direction in real-time. The methodology for this project is divided into four major parts: video capture, eye detection, calibration, and gaze estimation.

3.1 Video Capture

The initial step in the process involves capturing real-time video using a webcam. The video frames are then processed in real-time for eye detection.

3.2 Eye Detection

Each captured frame is analyzed for detecting the pupil's position. An algorithm is applied that uses image processing techniques to identify the ellipse representing the pupil. The center of this ellipse is taken as the current position of the pupil.

3.3 Calibration

Calibration is required to accurately map the coordinates of the pupil position in the image to the actual gaze point on the screen. The user is asked to gaze at different points on the screen, and the corresponding pupil positions are recorded. This data is used to compute a transformation function between the pupil positions and the screen coordinates.

3.4 Gaze Estimation

The final step is gaze estimation, which is done by continuously capturing the pupil's position and applying the transformation function calculated in the calibration step. This estimates the point on the screen that the user is gazing at. The estimated gaze point is then displayed on the screen for real-time visualization.

4 IMPLEMENTATION DETAILS

The system was built using Python with the help of various libraries and modules including Pygame for displaying visual stimuli and capturing user interaction, OpenCV for video capture and image processing, and TensorFlow for running the DeepVOG model.

4.1 Head-Chin Rest and Camera Setup

A custom-built head-chin rest (figure 2) is used to ensure the user's head remains steady during operation, which is crucial for accurate pupil detection and gaze estimation. The head-chin rest is constructed from PVC pipes, screws, and knobs, with a wooden base made from a 2 by 4 plank. The entire assembly is attached to the desk using a clamp, which provides stability. Epoxy glue was used to secure the elements of the structure. The head rest, chin rest, and camera positions are adjustable, allowing the system to adapt to different users and conditions. The camera is fixed to the head rest using a small adjustable arm made from a PVC pipe, maintaining a consistent image capture angle and distance.

4.2 Eye Detection Models

The system implements two different models for pupil detection – the DeepVOG model and Pupil Labs' Detector2D model. The DeepVOG model is a deep learning model specifically trained for gaze estimation, while the Detector2D model utilizes conventional computer vision techniques for pupil detection. The user can switch between these two models during operation, providing flexibility and adaptability to different conditions and requirements.

4.3 Real-time Operation and Calibration

The system operates in real time, processing 30 frames per second. Calibration is performed by asking the user to focus on specific points displayed on the screen. The system records the pupil position corresponding to these points and calculates a transformation function that maps pupil positions to screen coordinates. This

Fig. 2. Adjustable head-chin rest with camera setup. The head rest, chin rest, and camera positions are adjustable for different users and conditions.

calibration is performed every 60 frames (approximately every 2 seconds), to account for any possible changes in lighting conditions or slight movements of the user.

4.4 Mean Absolute Error (MAE) Calculation

The system includes an option to calculate the Mean Absolute Error (MAE) of the gaze estimation. This is done by comparing the estimated gaze point with the actual point the user is instructed to look at. This feature provides a measure of the system's accuracy.

4.5 Visual Feedback

The system provides real-time visual feedback by displaying the estimated gaze point on the screen. Additionally, it includes a 'rolling ball' feature, where a ball moves across the screen and the user is instructed to follow it with their eyes. This feature is useful for testing the system and observing the accuracy of the gaze tracking.

5 EVALUATION OF RESULTS

Upon successfully calibrating the system using the DeepVOG model, we conducted several evaluation experiments to assess the accuracy of our gaze tracking implementation. In each experiment, a red dot was displayed at a random location on the screen, and the user was instructed to focus their gaze on it. We then recorded the predicted gaze locations over a period corresponding to 30 frames, and computed the mean absolute error (MAE) between these predicted gaze locations and the actual location of the red dot. To increase the reliability of our results, the initial 20 percent of the gathered data, which could include user errors, was omitted from the MAE calculation.

Table 1. Evaluation Results

Dot Position (pixels)	**MAE (pixels)**	**MAE (inches)**	**MAE (degrees)**
(x:1583, y:971)	51.157	0.470	0.84
(x:174, y:72)	36.992	0.340	0.61
(x:1460, y:600)	33.727	0.310	0.55
(x:934, y:698)	29.368	0.270	0.48
(x:1551, y:524)	30.236	0.278	0.50
(x:503, y:150)	34.665	0.318	0.57
(x:763, y:64)	13.802	0.127	0.23
(x:372, y:1065)	42.847	0.394	0.7
(x:660, y:291),	16.512	0.152	0.27
(x:1434, y:171)	49.934	0.459	0.82
Mean	33.924	0.31	0.56

Note that we were unable to achieve a successful calibration using the Pupil Labs model, so all the results presented here are based on the DeepVOG model.

The distance from the screen to the eye was 32 inches, the screen had a pixel density of 108.85 PPI and the screen resolution was 2560x1440 pixels. Therefore, we have converted the MAE from pixels to inches using the known PPI for more intuitive understanding of the errors. We also calculated the MAE in degrees using the formula $\text{MAE}_{\text{degrees}} = \arctan(\text{MAE}_{\text{inches}}/32)$ for a comprehensive evaluation of the system's performance.

6 DISCUSSION OF BENEFITS AND LIMITATIONS

Our pupil tracking solution offers several benefits, but it also has limitations, which largely depend on the computational capabilities of the user's system.

6.1 Benefits

- **Flexibility:** Our approach can leverage two different models - the DeepVOG model for high accuracy and the Pupil Labs model for speed and resource efficiency. This flexibility allows users to choose the model that best fits their needs and system capabilities.
- **Calibration:** Our calibration process has been designed to be efficient and user-friendly, which enhances the accuracy of gaze tracking.
- **Hardware:** The adjustable head-chin rest we developed from inexpensive materials is extremely easy to use and sturdy.

6.2 Limitations

- **System Dependency:** The frame rate and overall performance of the eye-tracking system depend on the hardware capabilities of the user's system. While high-end systems can run the DeepVOG model with high frame rates, more

modest systems may experience performance degradation, with frame rates dropping to as low as 3-4 FPS. This could result in noticeable lag in gaze tracking, impacting the overall user experience.
- **Trade-off between Speed and Accuracy:** The Pupil Labs model offers high frame rates across a wide range of hardware, making it suitable for systems with limited resources. However, it is a lot less accurate than the DeepVOG model, presenting a trade-off between speed and accuracy.
- **Calibration Accuracy:** While our calibration process is generally reliable, low frame rates could potentially affect its accuracy. We recommend conducting the calibration process on a system with sufficient computational resources.

7 FUTURE WORK

While we take pride in the work we have accomplished with this project, we acknowledge there is ample room for growth and improvement. As a result, we have identified several avenues for future exploration:

- **Improved and more efficient ML models:** The current model we are using is resource-intensive and requires a powerful GPU. We plan to investigate more efficient and lighter machine learning models to enhance the performance of the gaze tracking system without such heavy resource requirements.
- **Research into alternative computer vision-based pupil detection methods:** We recognize that Pupil Labs may not provide optimal accuracy in our project. Therefore, we plan to investigate other computer vision techniques for pupil detection that could potentially enhance the accuracy and robustness of our gaze tracking system.
- **Integration with VR hardware:** Looking forward, we are keen on exploring the integration of our gaze tracking system directly into VR headsets. This could offer an additional level of convenience and user immersion.

Our humble project has laid a foundation upon which we hope to build in the future. We remain committed to learning, experimenting, and making meaningful contributions to this dynamic field.

8 CONCLUSION

In this project, we've taken on the challenge of implementing, integrating, and applying advanced machine learning and computer vision techniques to create an effective gaze tracking system in a VR context, all without the extensive resources often available to large corporations. It's clear that our efforts haven't broken new ground in the field of eye and gaze tracking technology, but we're proud to have demonstrated that resourcefulness and dedication can yield meaningful results.

We acknowledge our project as a small piece within a much larger puzzle. We are grateful for the substantial body of work that preceded ours, and we hope our modest effort will add to the shared knowledge in the field of VR eye and gaze tracking. We are enthusiastic about continuing to learn, adapt, and contribute within our capacity.

ACKNOWLEDGMENTS

We extend our deepest gratitude to Professor Douglas, John Akers and the TAs for their invaluable guidance and support.

REFERENCES

Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-Based Interaction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Seattle, Washington) *(UbiComp '14 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1151–1160. https://doi.org/10.1145/2638728.2641695

Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophey, Virginia L. Flanagin, Peter zu Eulenburg, and Seyed-Ahmad Ahmadi. 2019. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of Neuroscience Methods* 324 (2019), 108307. https://doi.org/10.1016/j.jneumeth.2019.05.016